|  |  |  |
| --- | --- | --- |
|  | **REPUBLIQUE TUNISIENNE**  \*\*\*\*\*  MINISTERE DE L'ENSEIGNEMENT  SUPERIEUR ET DE LA  RECHERCHE SCIENTIFIQUE  \*\*\*\*\*  **Département : Ingénierie**  **Informatique** | Image associée |

**RAPPORT**

**De**

**Projet de spécialité**

Analyse des journaux avec Hive

**Élaboré par :** Habib AROUA

Anas NAJJAR

Imen TRABELSI

Manel TRABELSI

**Encadré par** : Mr. Aymen HAJ KACEM

**Année Universitaire : 2018 / 2019**

|  |
| --- |
| ***Remerciement***  *Premièrement nous remercions Dieu source de toute connaissance...*  *Nous tenons à présenter nos vifs remerciements et nos profondes gratitudes à notre encadrant monsieur Aymen HAJ KACEM pour sa pédagogie, sa compétence, sa modestie et son aide précieuse tout au long de ce projet.* |

**Sommaire**

# **Introduction Générale**

De nos jours les sources d'informations sont considères comme une grande richesse dans l'entreprise mais qui sont volumineuses, déstructurées et hétérogènes d'où le besoin d'analyser ces derniers ans d'obtenir une vision globale sur l'entreprise.

Les systèmes d'aide à décision (Business intelligence en Anglais) répondent à ces besoins en proposant des modèles et des techniques de manipulation des données à aider les dirigeants de l'entreprise à prendre les bonnes décisions dans les plus brefs délais. En effet, en collectant des informations et en les traitant, il va diminuer le niveau d'incertitude autant que possible. Le système d'information doit permettre d'avoir une connaissance précise de l'environnement et de la situation interne an que les dirigeants choisissent les meilleures orientations stratégiques.

C’est dans ce cadre que se situe notre projet de spécialité dont le bit d’analyser les journaux en utilisant le structure de Hive.

Ce projet facilite la tâche de responsable des sondage en introduisant une manière informatisé pour collecter les informations nécessaires et les stocker dans une zone de stockage et les traiter avec requête HiveQl.

Notre projet a réalisé par une séquence de phases dont la projection se situe dans les cinq chapitres qui composent ce rapport.

Le premier chapitre présente le contexte général de ce projet. Quant au deuxième chapitre, il est dédié pour présenter le Hive, le troisième chapitre permet d’analyser le cahier de charge ei dégager les exigences, concernant le cinquième chapitre, il détaille la réalisation de notre outil décisionnel et décrit les différentes étapes de sa mise en place.

Ce rapport est clôturé par une conclusion générale qui présente une synthèse du travail réalisé, et ouvre des perspectives pour des éventuelles améliorations.

**Chapitre N°1 : Cadre général du projet**

**Introduction**

Ce chapitre d’inauguration vise à situer le présent projet dans son contexte général, à travers la description de la problématique incitée par son actuelle conjoncture fonctionnelle, et l’exposition des objectifs qu’on se propose d’atteindre par la réalisation de ce projet.

**Problématique**

L’augmentation du nombre des journaux électroniques durant les dernières années a entraîné une croissance en quantité et en qualité au niveau des données qui les concernent et qui sont utilisées dans les différents traitements.

Cette croissance de données, rendant la tâche des responsables de sondage (gestionnaires) plus difficile, impose une modernisation du système d’information. Ceci passe par la collecte des données éparpillées dans les différents systèmes opérationnels, et leur transformation en informations susceptibles de fournir une idée claire aux gestionnaires.

Malgré son caractère officiel et sa valeur fonctionnelle, ce document n’est pas élaboré d’une façon optimale, mais plutôt par une collecte manuelle des données depuis des sources hétérogènes, ce qui engendre une perte de temps remarquable et affecte la fiabilité des résultats présentés.

**Objectif**

**Conclusion**

Ce premier chapitre nous a permis de positionner dans notre projet dans son cadre général et de poser la problématique, qui nous guidera dans l’analyse du contexte fonctionnel existant et la conception de la solution dans les prochains chapitres.

**Chapitre N°2 : État de l’art**

**Introduction**

Aujourd’hui, on entend beaucoup parler des technologies Big Data : les chefs de projets en parlent et souhaitent expérimenter l’apport de ces technologies en termes de scalabilité, les commerciaux parlent de missions Big Data et de DataLab chez les clients, les RH cherchent des experts Big Data et des développeurs Hadoop qu’ils n’arrivent pas facilement à trouver. Un nombre très important de frameworks Big Data a vu le jour ces dernières années et l’écosystème Big Data est en pleine effervescence. Cependant, compte tenu du manque de maturité de son écosystème, plusieurs frameworks disparaissent à cause de leur complexité ou non adéquation avec les nouveaux besoins.

Nous allons, à travers cet article, introduire **Apache Hive,** un framework Big Data pour l’analyse des données. **Son utilité**: proposer une **abstraction** en dessus de MapReduce pour faciliter l’analyse de gros volumes de données. **Ses atouts** : de nombreux projets en production, une communauté active et un rythme de release assurant la compatibilité avec les nouvelles versions de Hadoop.

### **C’est quoi, HIVE ?**

[**Apache Hive**](http://hive.apache.org/) est un datawarehouse pour Hadoop. Il a été créé par Facebook pour devenir par la suite un**projet Apache open source**. Il ne s’agit pas d’une base de données relationnelle ni d’un datawarehouse classique.

**Si Hive n’est pas une base de données ni un datawarehouse, qu’est-ce donc alors ?**

Il s’agit d’un système qui maintient des métadonnées décrivant les données stockées dans HDFS. Il utilise une base de données relationnelle appelée **metastore** (Derby par défaut) pour assurer la **persistance des métadonnées.** Ainsi, une table dans Hive est composée essentiellement :

* D’un schéma stocké dans le metastore,
* De données stockées dans HDFS.

Avec les données du metastore, **Hive permet de manipuler les données comme si elles étaient persistées dans des tables** (au sens d’un système de gestion de base de données classique) et de les interroger avec son langage HiveQL.

Hive permet de convertir les requêtes HiveQL en jobs MapReduce ou Tez (à partir de la version 0.13 de Hive, une requête HiveQL peut être traduite en un job exécutable sur Apache Tez, qui est un framework d’exécution sur Hadoop pouvant remplacer MapReduce).

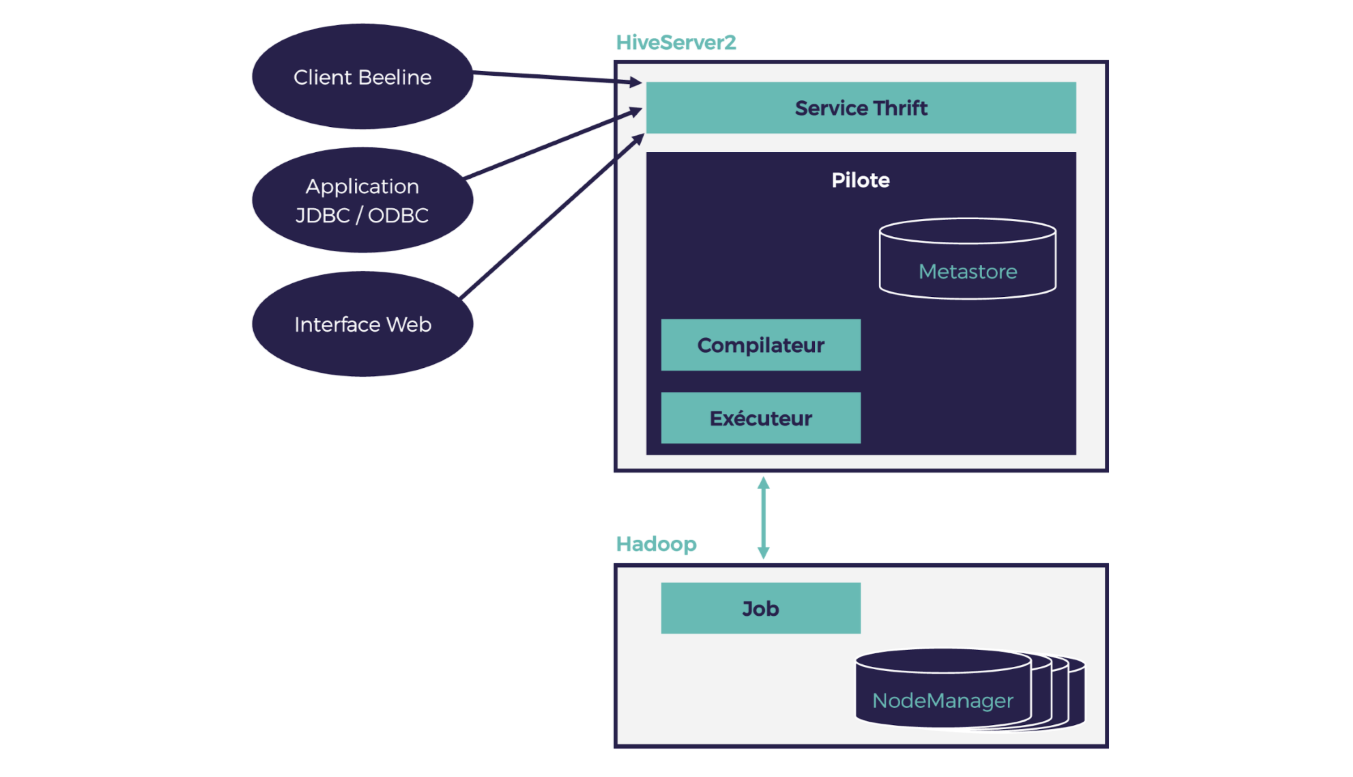
Ainsi, des profils familiers avec SQL (analystes, data scientists, etc.) n’ayant pas un background de développeur pourront écrire leurs requêtes HiveQL pour exploiter les données stockées dans HDFS sans se soucier de la partie programmatique de jobs.

D’un point de vue performance, Hive n’est certainement pas conçu dans une vision d’amélioration des performances d’exécution des jobs. En effet, les requêtes HiveQL (et donc job MapReduce ou Tez derrière) ne sont pas exécutées en temps réel et peuvent prendre quelques minutes ou quelques heures pour être exécutées. Ainsi, **l’avantage principal de Hive reste sa capacité d’abstraction par rapport à MapReduce.**

## **Architecture de HIVE**

Pour illustrer le fonctionnement de Hive et son architecture, nous allons décortiquer **l’exécution d’une requête Hive**. En effet, l’interaction Hive/Hadoop s’effectue selon les trois étapes suivantes :

1. **Envoi de la requête HiveQL** : en utilisant un client Hive (le client shell, un client JDBC/ODBC ou une interface web), la requête est envoyée au serveur Hive,
2. **Planification de la requête** : la requête est reçue par le driver (pilote). Elle est compilée, optimisée et planifiée comme un job,
3. **Exécution du job** : le job est exécuté sur le cluster Hadoop.



**Conclusion**

Nous avons introduit à travers cet article Apache Hive, son architecture et son langage de requête HiveQL qui est très similaire à SQL.

Ce **framework apporte une grande facilité** pour l’interrogation des données stockées dans HDFS en faisant une abstraction par rapport à MapReduce. Grâce à HiveQL, l’analyse des gros volumes de données devient aussi simple que le requêtage d’une base de données relationnelle avec SQL.

Les avantages de Hive par rapport aux autres frameworks d’analyse de données Big Data sont principalement : sa maturité, la communauté active qui l’utilise, ainsi que sa compatibilité avec les nouvelles versions de Hadoop.

Hive propose d’autres fonctionnalités plus avancées comme le tri et les jointures (plusieurs types de jointures comme le Shuffle Join, le Broadcast Join et le SMB Join sont définies dans Hive). Hive est aussi flexible grâce aux UDF (User Defined Function) qui sont des fonctions définies par l’utilisateur permettant d’étendre le langage et pouvant être ré-utilisées comme s’il s’agissait d’une bibliothèque externe.

Un second article sur le même sujet sera dédié à la présentation et l’utilisation des fonctionnalités avancées de Hive.

**Chapitre N°3 : Analyse**

**Introduction**

**Étude de l’existant**

**Description de l'existant**

**Critique de l'existant**

**Les exigences**

**Les exigences fonctionnelles**

**Les exigences non fonctionnelles**

**Diagramme de cas d’utilisation**

**Conclusion**